Deep Q-Network for 5G NR Downlink Scheduling

Walaa AlQwider, Talha Faizur Rahman, and Vuk Marojevic
Dept. of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS
{wq27|tfr42|vm602}@msstate.edu

2022 IEEE International Conference on Communications Workshops (ICC Workshops) | 978-1-6654-2671-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCWORKSHOPS53468.2022.9814547

***Abstract*—The Third Generation Partnership Project (3GPP) introduced the fifth generation new radio (5G NR) specifications which offer much higher flexibility than legacy cellular communications standards to better handle the heterogeneous service and performance requirements of the emerging use cases. This flexibility, however, makes the resources management more complex. This paper therefore designs a data driven resource allocation method based on the deep Q-network (DQN). The objective of the proposed model is to maximize the 5G NR cell throughput while providing a fair resource allocation across all users. Numerical results using a 3GPP compliant 5G NR simulator demonstrate that the DQN scheduler better balances the cell throughput and user fairness than existing schedulers.**

***Index Terms*—5G, deep learning, reinforcement learning, Q-learning, resource allocation, scheduling.**

I. Introduction

New radio (NR) is the fifth generation (5G) cellular communications technology that offers many new features for enabling flexible communications services and achieving the 5G performance targets. Among the new features is the support of a multi-numerology structure, which is characterized by the subcarrier spacing (SCS) and the transmission time interval (TTI) [1]. This flexibility enhancement has two sides: On the one hand, it is the enabler for enhanced mobile broadband (eMBB), ultra-reliable and low latency communications (URLLC), and massive machine-type communications (mMTC). On the other hand, it makes the system and resource management more complex and, therefore, only one or a few modes may be initially supported by a network.

Radio resource management (RRM) in NR is responsible for allocating, managing, and orchestrating resource blocks (RBs). State-of-the-art RRM schemes, such as proportional fair [2], round-robin [3], and channel-dependent scheduling [4], focus on design goals that revolve around network-oriented objectives, such as fairness or throughput, and making them insensitive to time-sensitive applications. Other scheduling strategies may be designed based on the human's understanding of the network. These cannot react to the rapid changes in the network dynamics, including wireless channel statistics, user mobility patterns, instantaneous radio resource availability, and traffic load variability [5].

Schedulers commonly consider the buffer state information and the channel quality indicator (CQI) for optimizing the allocation of radio resources to user equipment (UEs). This corresponds to an optimal control problem of a Markov decision process (MDP) that requires reinforcement learning (RL) [6]. In particular, deep RL (DRL) has shown tremendous potential in solving complex problems by approximating the interactions between the resource allocation decisions and different performance metrics. It is capable of learning the best policy for a network [6]. Ye, et al. [7] leverage DRL for resource allocation in a vehicle-to-vehicle (V2V) communications context in order to minimize the interference of the V2V links to the vehicle-to-infrastructure links while respecting the latency constraints of V2V communications. Zheng, et al. [8] propose a single agent DRL algorithm to address the problem of channel assignment for hybrid non-orthogonal multiple access-based cellular networks. Similarly, [9] introduce a framework that combines multiple scheduling rules to meet the quality of service requirements in terms of packet delay and packet delivery ratio.

Classic RL algorithms have inevitably high computational complexity and are only applicable to problems with small state-action spaces. In this regard, the deep Q-network (DQN) is considered where the neural network (NN) is applied to approximate the state-action value function. A DQN model is developed in [10] where a number of networks using different MAC protocols try to access the time slots of a common wireless medium. The DRL agent learns by interacting with users employing different channel access mechanisms and learns to transmit in those slots where other users are idle.

In this paper we leverage the benefits of machine learning for solving the complex RRM problem of 5G NR by designing a DQN-based downlink scheduler. The goal of the proposed scheduler is to solve the dynamic resources allocation problem and optimize the cell throughout while allocating a fair amount of radio resources to the UEs requesting service. We evaluate the performance of the proposed scheduler against state-of-the-art benchmark schedulers. The results show that our DQN design makes best use of the available resources to achieve a very high cell throughput without sacrificing user fairness.

While we consider a system model that is compliant with the Third Generation Partnership Project (3GPP) Release 16 [11], the proposed strategy is data driven which makes it an ideal candidate for the open radio access network (O-RAN) architecture. O-RAN, which is standardized by the O-RAN Alliance [12], is transforming the RAN industry by enabling open, intelligent, virtualized, and interoperable RAN implementation and operation. O-RAN provisions for the integration of artificial intelligence (AI) for its xApps and rApps that implement the near real-time and non-real-time RAN intelligent controllers, respectively. Specifically, the open and virtualized architecture facilitates the development of a variety of xAPPs for near-real-time control, including AI-enhanced schedulers, that can be integrated into operational 5G and Beyond 5G networks for offering differentiated network services [13].

The rest of the paper is organized as follows: Section II introduces the system model and problem formulation. Section III describes the design of the proposed DQN-based scheduling for 5G NR. Section IV provides the performance evaluation and Section V draws the conclusions.

II. System Model and Problem Formulation

A. 5G NR Resource Allocation

The 5G NR carrier can be of different bandwidths up to 400 MHz. In order to be able to serve UEs that can only handle a subset of the carrier bandwidth, 5G introduces the bandwidth part (BWP) as a subset of the carrier bandwidth.

The smallest resource to be allocated is the resource block (RB) which contains 12 consecutive resource elements across the subcarrier space within one OFDM symbol, irrespective of the SCS. Different RBs can be allocated to different UEs in a cell within the TTI and reallocated across TTIs; each RB is allocated to only one UE per TTI.

The 5G NR downlink supports two resources allocation types: Type 0 and Type 1. Type 0 uses a bitmap to indicate the resources block groups (RBGs) which are allocated to the UEs. The RBG combines a certain number of consecutive RBs to be assigned to a single UE. Type 1 assigns contiguous virtual RBs UEs. The network encodes the starting virtual resource block (RB_{start}) and the length of the contiguously allocated RBs (L_{RBs}) in the *resource indication value* and includes this field in the downlink control channel. While Type 0 incurs more control overhead than Type 1 to inform the UE about the allocated RBs, it is more flexible than Type 1.

Because of its higher flexibility we consider the resource allocation Type 0 where the BWP is divided into RBGs whose number is determined by the following formula [11]:

$$N_{RBG} = \lceil (N_{BWP}^{size} + \mod (N_{BWP}^{start}, P))/P \rceil . \quad (1)$$

Parameter N_{RBG} captures the number of RBGs in a BWP of size N_{BWP}^{size} RBs, N_{BWP}^{start} is the RB index indicating the start of the BWP, and P is the size of the RBG and is determined by the higher layer parameter rbg-Size and N_{BWP}^{size} [11].

In each TTI, the 5G base station (gNB) allocates a RBG to a particular user for downlink data transfer, assuming there is data to be transferred. The user is determined by the allocation strategy and the performance metric to be optimized, such as throughput or resource fairness. This centralized resource allocation method takes certain inputs, including the CQI, the buffer status, and the served data rate from each UE. The scheduler may process some or all of this information to make its decision.

B. Performance Metrics

We consider the downlink transmission where N active UEs are requesting to be served by the gNB as shown in Fig. 1. These UEs are requesting packets from the gNB, and the gNB performs the RBG allocation in each TTI. There are N_{RBG} RBGs, RBG^1 to $RBG^{N_{RBG}}$. Our objective is to effectively balance between throughput and fairness which is why the performance metrics are the cell throughput and the resource allocation fairness among the active UEs.

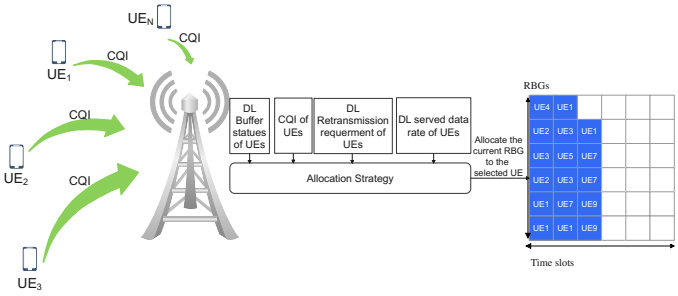


Fig. 1: General radio resources allocation model.

The achievable throughput d in Mbps for the n-th UE on RBG^i is given by the following formula assuming we have one carrier [14]:

$$d_{UE_n,i} = V_{layer,UE_n} \cdot Q^M_{UE_n,i} \cdot f \cdot R_{i,UE_n} \cdot \frac{RBG^i_{size} \cdot 12}{TTI_{dur}} \cdot (1 - OH) \cdot 10^{-6} \quad [Mbps]. \quad (2)$$

Parameter V_{layer,UE_n} indicates the number of layers, $Q^M_{UE_n,i}$ the modulation order, and R_{i,UE_n} the code rate efficiency on RBG^i. The values for $Q^M_{UE_n,i}$ and R_{i,UE_n} are obtained from the modulation and coding scheme, which depends on the channel quality for the scheduled UE in the given TTI. Parameter f is a scaling factor which does not depend on the scheduled UE. Symbol RBG^i_{size} indicates the number of RBs in RBG^i, OH captures the control channel overhead, which is determined by the direction of the transmission (uplink or downlink) and the frequency range (FR1 or FR2), and TTI_{dur} indicates the duration of the TTI in seconds and depends on the SCS.

The achievable data rate per TTI can be formulated as

$$D(TTI) = \sum_{i=1}^{N_{RBG}} D_i(TTI) \times C_i(TTI), \quad (3)$$

where $D_i(TTI)$ is of size $1 \times N$ and captures the achievable data rates of the N UEs if they were individually scheduled on the current RBG in the current TTI, and $C_i(TTI)$ is a binary vector of size $N \times 1$ indicating the scheduled user to receive data on RBG^i.

As the fairness metric, we compute the Jain fairness index (JFI) [15]:

$$JFI = \frac{(\sum_{n \in N} x_n)^2}{N \sum_{n \in N} x_n^2}. \quad (4)$$

Parameter x_n is the fraction of RBs assigned to UE_n. The maximum fairness (JFI = 1) is achieved when each UE gets the same share of RBs. If the scheduler allocate most of the RBs to a small subset of UEs, the JFI will be low, or close to zero, and this indicates that the RBs are not allocated fairly among the UEs.

C. Problem Formulation

The objective of the scheduler is to assign UEs to RBGs in such a way to maximize the total achievable data rate in the scheduled TTI across all RBGs while achieving high users fairness in accessing the available resources. Hence, it becomes a joint optimization problem:

$$\max \{D, JFI\}, \tag{5}$$

$$s.t. \sum_{n=1}^{N} c_{i,n}(TTI) \leq 1, \forall i \in [1, 2, ..., N_{RBG}]. \tag{6}$$

D is the accumulated or average cell throughput and JFI the resulting JFI. Constraint (6) ensures RBG exclusivity in each TTI. That is, at most one UE can be selected for each RBG^i per TTI.

III. DQN for Resource Allocation in 5G NR

DRL is a learning scheme for sequential decision problems with the aim of maximizing a cumulative future reward. It contains two important components, the agents and the environment, where agents use deep NN (DNNs) to learn by interacting with the environment [16]. The environment is modeled as a MDP which provides the mathematical framework for modeling decision making problems whose outcome is random and controlled by a decision maker, or agent.

We start by modeling the problem as a MDP and define the state and action spaces, and the reward function. Then we introduce the DQN agent. Finally, we define the agent training and the DQN agent engagement for 5G resources allocation.

The scheduler operates on the basis of a TTI. Hence, the throughput and fairness are evaluated in each TTI. We drop *TTI* from the symbols used here for improving the readability

A. MDP for Modeling the Environment

Each RBG is characterized by states based on the achievable data rate and the fairness indicator for each UE. Therefore, we group the UE data rates and the associated fairness indicators in a unique state vector as,

$$s_i = [d_{UE_1,i}, ..., d_{UE_N,i}, PF_{UE_1,i}, ..., PF_{UE_N,i}], \tag{7}$$

where the proportional fairness (PF) indicator is given as,

$$PF_{UE_n,i} = \frac{d_{UE_n,i}}{ds_{UE_n}}, \tag{8}$$

and ds_{UE_n} is the historical served data rate of UE_n. Recall that i indexes the RBG as RBG^i.

The action space $\mathcal{A}$ are the UEs to be served:

$$\mathcal{A} = [UE_1, UE_2, ..., UE_N]. \tag{9}$$

The reward function guides the actions and encourages the agent to learn about the best actions. Since our goal is to maximize cell throughput and user fairness, we define the reward function as,

$$r_i = d_{UEn,i} \times \frac{\min\limits_{UE_n} PF_{UE_n,i}}{\max\limits_{UE_n} PF_{UE_n,i}}. \tag{10}$$

This reward function discounts the expected data rate by decreasing the gap between the UEs associated with the minimum and maximum PF to maintain a high fairness by preventing the reservation of resources exclusively to the UE that has the best channel state. Here we do not consider punishment because all UEs are eligible for being scheduled and have data in their buffers.

B. DQN Agent

After formalizing the problem as a MDP, next we need to choose the algorithm to implement the agent. Q-learning [16] is a method that can be used to implement the agent side of the RL system. It has the ability to solve dynamic decision making problems by finding good policies and choosing the actions that maximize the accumulative reward function without requiring prior knowledge about the system model. Since we have a continuous state space and a discrete action space, Q-learning is the method that best fits this problem.

The Q-values for all state-action pairs $\{s, a\}$ are presented in a lookup table and tell the agent how good it is to be in state s and take action a. The Q-values are randomly initialized and updated in each iteration until they converge to the optimal Q^*.

For our problem, we define step $t \in [1, 2, ..., N_{RBG}]$, which corresponds to i that is used to index the RBGs, whereas episodes correspond to the TTIs. Since the TTI is the scheduling interval in 5G NR, the proposed trained DQN method allocates all RBGs in a TTI in real time, before moving on to the next TTI.

A drawback of Q-learning is scalability when the state and the action spaces become prohibitively large. For large state and action space problems, it is infeasible to keep track of each $\{s, a\}$-pair. In order to overcome this limitation, the DQN has been developed where DNNs are considered to approximate the Q-values, completely removing the lookup table. Such a DQN agent consists of three main parts: the main network, the target network, and the replay memory $\mathcal{D}$. At each time t, the main network takes the observed state s_t from the environment and outputs $Q(s_t, a_t|\theta)$ for each action a_t in $\mathcal{A}$, where θ represents the training parameters, or weights, of the DNN. Then the action is selected using the ϵ-greedy algorithm where the agent selects the action $a_t = \arg\max_{\acute{a} \in \mathcal{A}} Q(s, \acute{a})$ with probability $1 - \epsilon$ or a random action with probability ϵ. This ϵ-greedy algorithm helps the agent balancing between exploration and exploitation to avoid ending up in a local minimum. After choosing the action, the reward function r_t is calculated, the next state s_{t+1} is observed, and the resultant tuple (s_t, a_t, r_t, s_{t+1}) is added to $\mathcal{D}$.

In each training iteration, the agent randomly samples mini-batches from $\mathcal{D}$ to train the DNN. This way, the temporal correlation between the training samples can be broken and this is the main purpose of using the replay memory. The DQN is optimized by updating the weights θ. This is an iterative process that minimizes the mean square error between the main DNN and the target DNN estimations. This error is

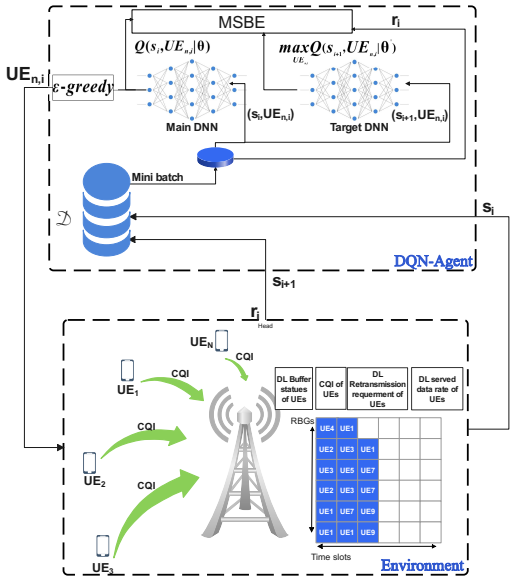


Fig. 2: The training stage of the DQN based RBG allocation.

known as the mean-squared Bellman error (MSBE) [16]:

$$L(\theta_t) = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}\in\mathcal{D}}[r_t(s_t,a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1},a_t|\acute{\theta}) - Q(s_t,a_t|\theta)]^2. \quad (11)$$

Parameter $\acute{\theta}$ is the weight of the target DNN which is a delayed version of θ. The target DNN is an identical copy of the main DNN and is used to calculate the target Q-values. Employing a target DNN enhances the stability of the learning process because the target is calculated from a more mature network whose weights are periodically updated every T steps by a smoothing update approach,

$$\acute{\theta} = \beta\theta + (1-\beta)\acute{\theta}, \quad (12)$$

where β is the small, real-valued smoothing parameter. The main DNN weights are updated using the stochastic gradient descent (SGD) method,

$$\theta = \theta + \frac{\alpha}{M}(TD)\nabla_\theta Q(s,a,\theta), \quad (13)$$

where TD is the temporal difference error between the outputs of the target and main DNNs,

$$TD = Q(s,a,\theta) - Q^{target}(s,a,\acute{\theta}). \quad (14)$$

C. DQN Based RBG Allocation

The DQN consists of two stages, the training stage and the testing stage. Fig. 2 illustrates the training stage which Algorithm 1 summarizes. During this stage, the DQN agent is trained for a number of episodes while interacting with the environment until the accumulated reward converges. One episode is one TTI during which the agent and the environment interact until the allocation of all available RBGs has been completed. The DNN weights and the replay memory content are transferred to the next episode. At the end of the training stage, the trained agent is saved and engages in the MAC layer of the gNB for resource allocation as per Algorithm 2.

Algorithm 1 DQN based RBG allocation training stage

Input: DQN structures, Environment simulator;
Start:
Initialize θ and replay memory $\mathcal{D}$;
for each episode (i.e. TTI) **do**
  for step t = i = 1 to N_{RBG} **do**
    Observe s_i using (2), (8), and (7);
    Forward s_i to the main DNN and get select a_i using ϵ-greedy;
    Calculate r_i using (9);
    Observe s_{i+1} using (2), (8), and (7);
    Store the sample (s_i,a_i,r_i,s_{i+1}) in $\mathcal{D}$;
    Sample mini-batch from $\mathcal{D}$ optimize θ using (11), (13), and (14);
    Update $\acute{\theta}$ every T steps using (12);
  end for
end for
Output: Trained DQN agent

Algorithm 2 DQN based RBG allocation test stage

Input: Trained DQN agent, Environment simulator;
Start:
for each TTI **do**
  for each RBG^i **do**
    Observe s_i using (2), (8) and (7);
    Forward s_i to the trained DQN agent and select a_i using greedy method;
    Calculate r_i using (9);
    Selected UE = a_i;
    Allocate RBG^i to the selected UE;
  end for
end for

IV. Performance Evaluation

We evaluate the proposed scheduler numerically against state-of-the-art scheduling techniques using 5G NR compliant system parameters of 3GPP Release 16.

A. Simulation Parameters

The DQN related parameters and the simulation parameters are summarized in Tables I and II, respectively. We design a DNN with three fully connected layers of 100 neurons in each layer and the $relu$ activation function. The DQN agent is trained for 1000 episodes, where each episode has 100 steps, i.e 100 RBGs, and employ the adaptive moment estimation method (Adam) for training. We consider a single gNB and 20 associated UEs. The UEs experience different channels and have different downlink packet sizes, but the periodicity of the packet generation is the same for all UEs. The channels between the gNB and each of the UEs are changing over time.

The simulations are performed in MATLAB using the 5G Toolbox and the *NR TDD Symbol Based Scheduling Performance Evaluation* example. The RL Toolbox is leveraged to design the DQN agent and the MDP environment.

The benchmark schedulers are:

- Round-robin (RR): Each UEs is allocated RBGs, one UE after another, regardless of the channel.
- Best CQI (BestCQI): Each RBG is allocated to the UE that has the best CQI among all UEs in the current TTI.

TABLE I: DQN hyperparameters.

Parameter	Value
Number of episodes	1000
γ	0.99
$\mathcal{D}size$	10^6
Initial ϵ	0.99
ϵ decay	0.0001
min ϵ	0.001
β	0.001
T	10
Training rate α	0.001
Mini-Batch size M	32

TABLE II: 5G NR system simulation parameters.

Parameter	Value
Bandwidth	20 MHz
SCS (N_{RB})	15 kHz (100), 30 kHz (50)
P (SCS)	8 (15 kHz), 4 (30 kHz)
No. of UEs N	20
TTI (SCS)	1 ms (15 kHz), 0.5 ms (30 kHz)
CQI periodicity	10 ms
Simulation time	100 radio frames

- Proportional Fair (PF): RGBs are allocated to UEs balancing cell throughput and user fairness according to (8).

B. Results and Analysis

We consider three performance metrics: throughput (2)–(3), fairness (6), and cell goodput, which is the amount of downlink data that is successfully transferred from the gNB to the UEs and is calculated by subtracting the unacknowledged packets from the sent packets.

Fig. 3 plots the reward over the episode to illustrate convergence during the training process. The average reward and the episode reward converge after fewer than 400 episodes.

The simulation time is 100 radio frames, i.e. 1 s. Figs. 4 plots the cell throughput and 5 captures the throughput distribution. Fig. 4 shows the superiority of BestCQI over the other schedulers. Fig. 5 reveals that BestCQI does not schedule all UEs in 100 radio frames and the median throughput is much lower than that of the other schedulers. This figure indicates the high variance of the achieved UE throughput values when employing BestCQI. The proposed DQN, PF and RR show a fair distribution of achieved throughput values among the UEs where each UEs gets scheduled. The proposed DQN-based allocation scheme considerably outperforms PF and RR for both SCSs. BestCQI outperforms the DQN but at the cost of starving some of the UEs. The BestCQI scheduler tends to starve the users who report a poor channel quality as it always prioritizes the UEs experiencing good channels and enabling higher bit loading.

As mentioned before, the goodput is the amount of data received correctly by the receivers. Figs. 6 and 7 provide the goodput results. We observe the same behaviors as for the throughput and conclude that our proposed schemes performs much better than PF and RR, irrespective of the SCS. It achieves lower cell throughput performance than BestCQI but

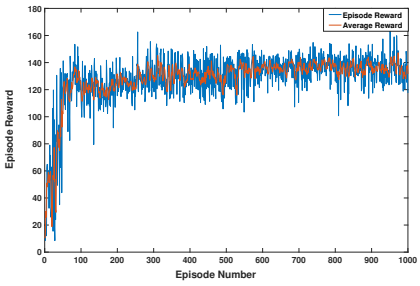
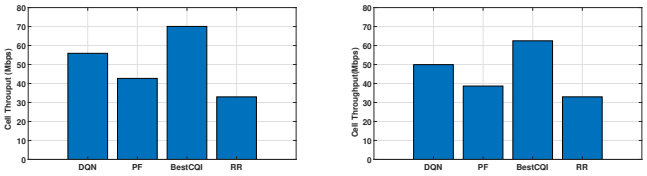
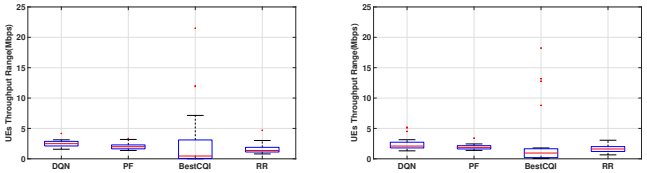


Fig. 3: DQN training convergence.



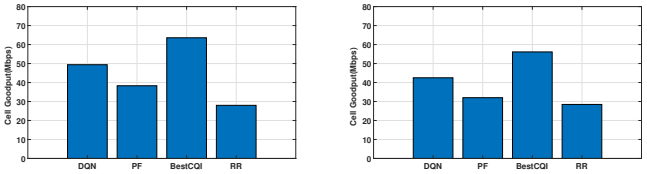
(a) SCS = 15 kHz (b) SCS = 30 kHz

Fig. 4: Cell throughput over 100 radio frames.



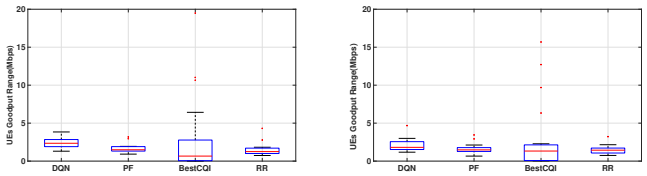
(a) SCS = 15kHz (b) SCS = 30kHz

Fig. 5: UEs throughput over 100 radio frames.



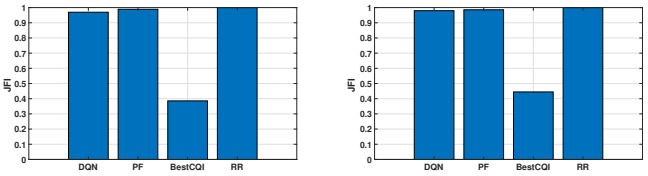
(a) SCS = 15kHz (b) SCS = 30kHz

Fig. 6: Cell goodput over 100 radio frames.



(a) SCS = 15kHz (b) SCS = 30kHz

Fig. 7: UEs goodput over 100 radio frames.



(a) SCS = 15kHz (b) SCS = 30kHz

Fig. 8: Jain Fairness Index (JFI) over 100 radio frames.

all the UEs are served during a radio frame and the median throughput is the highest for the DQN scheduler.

For the fairness comparison, we calculate the JFI of the accumulated resources allocated to the different UEs over the simulation time of one radio frame. Fig. 8 shows the JFI for both SCSs. As expected, RR has the best JFI because it schedules UEs in cycles, offering each UEs an equal share of resource. The proposed method performs considerably better than BestCQI and it is very close to the PF scheduler, which emphasizes both fairness and throughput, just like the DQN. As expected, the BestCQI has a very low JFI compared to the other schedulers.

These results show the effectiveness of the proposed DQN scheduler that implements a new way of resource allocation in 5G NR. It balances two competing objectives—cell throughput/goodput and user fairness—and performs very close to methods tailored to one or the other objective. While the PF has been designed to balance the same objectives, its performance is inferior to that of the proposed DQN design for 5G NR downlink scheduling

C. Computational Complexity

We start by calculating the complexity of the training, according to Algorithm 1. In each step, the computational complexity order is $O(F)$, where F is according to the DNN structure and is found to be $F \triangleq 2Nd_1+\sum_{g=1}^{G-1} d_g d_{g+1}+d_G N$ [17], where G is the number of hidden layers, d_g is the number of neurons in hidden layer g, and $2N$ and N are the dimensions of the input and output layers, respectively. As the operation is repeated across steps and episodes, the overall complexity of the training phase becomes $O(FMN_{RBG})$, where M symbolizes the number of episodes and N_{RBG} the number of steps per episode. After training, the DQN scheduler performs FN_{RBG} operations every TTI. This computational complexity is affordable with current computing technology as opposed to an exhaustive search scheduler whose complexity is $O(N^{N_{RBG}})$ per TTI.

The complexity of the DQN allocation stage according to Algorithm 2 is $O(F)$ for each RBG in a given TTI. The corresponding complexity of the PF scheduler is $O(N)$ as is the complexity of the BestCQI approach because both evaluate all N UEs for each RBG, unlike the RR scheduler which is characterized by $O(1)$.

V. Conclusions

In this paper we have proposed, designed, and numerically analyzed a DQN-based scheduler for the 5G NR downlink. Using DNNs and training data, the propose scheduler balances the maximization of the cell throughput and fairness among the UEs. The simulation results show that the proposed data driven scheduler learns quickly and outperforms benchmark methods by best balancing the two performance metrics. The main advantage of the proposed DQN scheduler is its flexibility, where the reward function can be customized to the radio environment and other performance indicators. For example, designing a DQN scheduler that minimizes the radio frequency interference level in shared spectrum among active and passive users is an emerging research direction for next generation wireless. Another direction to extend this work is to add power control to the reward function and use multiple agents for different cells in order to manage the inter cell interference.

Acknowledgment

This work was supported in part by the National Science Foundation under grant numbers ECCS-2030291 and CNS-2120442.

References

[1] E. Dahlman and S. Parkvall, "NR – the new 5G radio-access technology," in *IEEE VTC Spring*, 3-6 June, 2018, pp. 1–5.

[2] R. Kwan, C. Leung, and J. Zhang, "Proportional Fair Multiuser Scheduling in LTE," *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 461–464, 2009.

[3] O. Østerbø, "Scheduling and capacity estimation in LTE," in *2011 23rd International Teletraffic Congress (ITC)*, 2011, pp. 63–70.

[4] W. Fang, G. Wang, G. B. Giannakis, Q. Liu, X. Wang, and H. Deng, "Channel-Dependent Scheduling in Wireless Energy Transfer for Mobile Devices," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3330–3340, 2020.

[5] H. B. Pasandi and T. Nadeem, "Challenges and limitations in automating the design of MAC protocols using machine-learning," in *1st ICAIIC*, 2019, pp. 107–112.

[6] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[7] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019.

[8] J. Zheng, X. Tang, X. Wei, H. Shen, and L. Zhao, "Channel assignment for hybrid NOMA systems with deep reinforcement learning," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1370–1374, 2021.

[9] I.-S. Comşa et al., "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Trans. Network and Service Management*, vol. 15, no. 4, pp. 1661–1675, 2018.

[10] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[11] The 3rd Generation Partnership Project (3GPP), "Physical layer procedures for data," Technical Specification (TS) 38.214, 07 2020, Version 16.2.0.

[12] O-RAN ALLIANCE, "O-RAN Architecture Description," Technical Specification (TS) O-RAN.WG1, 07 2021, Version 05.00.

[13] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, "Toward next generation open radio access network–what O-RAN can and cannot do!" *arXiv 2111.13754*, pp. 1–8, 2021.

[14] The 3rd Generation Partnership Project (3GPP), "User Equipment (UE) radio access capabilities," Technical Specification (TS) 38.306, 07 2020, Version 16.1.0.

[15] A. B. Sediq, R. H. Gohary, and H. Yanikomeroglu, "Optimal tradeoff between efficiency and Jain's fairness index in resource allocation," in *23rd IEEE PIMRC*, 2012, pp. 577–583.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[17] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2019.